Data Splash!
An Educational Game about Machine Learning

Andrew Gray

445348

Submitted to Swansea University in fulfilment
of the requirements for the Degree of Master of Science

Swansea University
Prifysgol Abertawe

Department of Computer Science

Swansea University

September 20th, 2020

Declaration

This work has not been previously accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed .... (candidate)

Date ....

Statement 1

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed .... (candidate)

Date ....

Statement 2

I hereby give my consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed .... (candidate)

Date ....

I would like to dedicate this work to the Hypnotoad.
All glory to the Hypnotoad.

Abstract

In your abstract you should aim to summarize the core contributions of your work in the context of the problem domain. Start by outlining the domain and the problems posed within it. Discuss how the methods you focus on approach the relevant problems. You should end your abstract by concretely stating the tangible outputs and deliverables you have created in order to complete your work on this document, and whether those outputs represent and improvement or alternative approach to existing methods.

Your abstract should be a couple or so paragraphs long, and roughly approximate the order and flow you then use for structuring the main document. If a viewer has read your abstract then they should already understand at a high level what it is you have created and delivered, and whether it is better than or comparable to existing methods. If your project is driven by a research hypothesis then the reader should know what that is at a high level from this section. Reading on, little should surprise the viewer.

For paper submission of your thesis you should physically sign your name on each of the above declaration statements and date them in black ink. For digital submissions you should sign and date them digitally using a touch or stylus input if available. There are pieces of software that allow you to write directly on PDF documents, or alternatively you can bring a signature into your document as a figure with a transparent or white background. If you do not have a stylus input / tablet like device you should ask your supervisor, as many in the department do their grading / work on digital tablets.

Acknowledgements

This is an opportunity to acknowledge and thank those who have supported you throughout your studies. Friends and colleagues who you have studied alongside, your families, and your mentors within the department are the usual suspects.

Contents

List of Tables **vii**

List of Figures **viii**

1 Introduction **1**

1.1 Motivations 1

1.2 Overview 2

1.3 Contributions 2

2 Background & Literature Review **3**

3 Methodology **5**

3.1 Overview of Application 5

3.2 Overview of Specific Game Components 6

3.3 Evaluation of Application 6

4 Implementation **9**

4.1 Tools 9

4.2 Frameworks 14

4.3 Packages Used 15

4.4 IDE Used 15

4.5 Intricacies of the Game Components 17

4.6 Example user stories (A UML term for case studies or example playthroughs) . 17

5 Conclusions and Future Work **19**

5.1 Contributions 19

5.2 Future Work 19

Bibliography 21

Appendices 23

A Implementation of a Relevant Algorithm 25

B Supplementary Data 27

List of Tables

List of Figures

4.1 A comparison between Java, Python and C++ to print an output to the console. [1] . 12

Chapter 1

Introduction

This document is intended both as a thesis template and a written tutorial on typesetting a professional looking academic document. The style of the template is designed to mimic an equivalent LaTeX document template that is commonly used for within the Computer Vision and Visual Analytics group here at Swansea. This LaTeX template is itself based on a LaTeX template named Custard.

1.1 Motivations

Large documents can become cumbersome to work with and format consistently. Sensibly chosen aesthetic cues are important to help imply structure and can greatly aid the reader in understanding your work. The accompanying LaTeX template uses abstraction to hide the formatting from the author during content preparation, allowing for consistent styling to be applied automatically during document compilation. In this Google Docs theme it is the responsibility of the author to manually adhere to the styling laid out in this template.

1.1.1 Objective

In this document we present a tutorial on thesis creation and typesetting, and discuss topics such as literature surveying and proper citation.

1.2 Overview

The remainder of chapter 1 outlines the document structure and the key contributions of this work is organized as follows. Chapter **??** reviews techniques for finding and properly citing external resources from the academic literature and online. In chapter **??** we show examples of how to typeset different types of content, such as internal references, figures, code listings, and tables. And lastly in chapter 5 we summarize the main contributions and key points to take away from this template.

1.3 Contributions

The main contributions of this work can be seen as follows:

- **A LaTeX thesis template**

  Modify this document by adding additional TeX files for your top level content chapters.

- **A typesetting guide of useful primitive elements**

  Use the building blocks within this template to typeset each part of your document. Aim to use simple and reusable elements to keep your LaTeX code neat and to make your document consistently styled throughout.

- **A review of how to find and cite external resources**

  We review techniques and resources for finding and properly citing resources from the prior academic literature and from online resources.

Chapter 2

Background & Literature Review

Chapter 3

Methodology

The university has subscriptions to a vast number of major academic journals spanning a wide range of subject areas. By accessing the internet from a university network connection (Eduroam or Ethernet), the paywalls of many journals will simply vanish without any need for login credentials.

3.1 Overview of Application

When you are working from outside of the university then connecting to an on campus machine via remote desktop (RemoteDesktopProtocol, TeamViewer, ect) or via port forwarding (ssh, ssh tunnel, ect) can allow you to access papers that would otherwise be behind a paywall.

If you do not have individual access to a machine that is exposed for ssh on the university network you can always use the computers in Linux Lab CF204[1] for the purpose of setting up an ssh port tunnel to proxy your internet through. These machines have fixed IPv4 addresses and respond to ssh using your student account credentials. While in use your internet will be routed[2] to the university and then out to the internet, granting you transparent access to journals without a paywall.

[1] One caveat of using computer lab machines for remote tunnelling is that a environmentally conscious student who has worked late in the computer lab might choose to switch off the machine you were using...

[2] Painfully slowly.

3.1.1 Design

3.2 Overview of Specific Game Components

The internet is big [2]. Knowing how to phrase a question to a search engine is therefore an invaluable skill. If the request is simple enough, even a poorly structured query will likely return usable results. For more difficult to find resources you can leverage the language of the search engine to gather relevant papers and resources for your research more efficiently.

```
https://www.gwern.net/Search
```

"Internet Search Tips" [3] provides an excellent review of methods and tips for scouring the internet for hard to find resources. You will also be less likely to get caught behind journal paywalls when working remotely without a tunnel as your queries can be made to look for raw pdfs that are often released by the authors directly.

3.2.1 Game Arena

3.2.2 Free Play

3.2.3 Learning Zone

3.2.4 Awards Zone

3.3 Evaluation of Application

BibTeX is a language for specifying resource citations. Every time you access and read an academic paper, take code from an online repository, or source the media such as images from existing works you should create a BibTeX entry in a file that you keep throughout your research. Software such as Mendeley [4] can help automate the process of building your BibTeX library of citations.

```
@INPROCEEDINGS{kaj86,
    author    = {Kajiya, James T.},
    title     = {The Rendering Equation},
    booktitle = {Proceedings of the 13th Annual Conference on Computer Graphics
        and Interactive Techniques},
    year      = {1986},
    series    = {SIGGRAPH '86},
    pages     = {143--150},
    address   = {New York, NY, USA},
    publisher = {ACM},
    isbn      = {0-89791-196-2},
    numpages  = {8},
    acmid     = {15902}
}
```

Listing 3.1: An example BibTeX entry for an academic paper published in conference proceedings [5].

The BibTeX code listing above (listing 3.1) shows an example of how to cite an academic paper, in this case one of the central papers in Computer Graphics research. The key **kaj86** is an arbitrary name chosen as a meaningful identifier for the resource. In the document text we can call on this resource as an inline citation using the LaTeX command `\cite{kaj86}` which produces [5] at the location it is called. As long as a citation has been used at least once somewhere within the document then a formatted full citation will be created in the bibliography at the end of the document with the same citation number that is shown inline.

It is considerably easier to be disciplined in methodically taking note of the resources you access and make use of as you access them, than it is to try and hunt them all down again at the time you need to write about them in your document. Invest time in being organized and consistent up front and it will be easier when you come to write up.

3.3.1 User Study

Chapter 4

Implementation

4.1 Tools

4.1.1 Programming Languages

For the implementation of our application, three primary programming languages deemed to be best suited for development. Apple's Swift programming language [6] got considered early on, due to the author's familiarisation with the programming language. The programming language gets used for creating applications for Apple's mobile and desktop operating systems, and with 1.5 billion [7] iOS devices in circulation, that was a lot of potential users. Additionally, Apple's iOS devices are prevalent within most educational settings, with Apple's iPad being one of the primary go-to devices. However, due to the language not supporting key frameworks required, or providing similar alternatives, the decision to not use this language got made.

We then got presented with three main options to use, **Python, R and JavaScript** along with HTML and CSS.

Python is a very popular programming language [8, 9], it is fast, easy-to-use, and easy-to-deploy programming language that gets widely used to develop scalable applications. Examples include YouTube, Instagram, Pinterest and SurveyMonkey [10]. The Python Software Foundation state that Python is a high-level object-orientated interpreted language with dynamic semantics. Due to the language being a high-level, it has many built-in data structures. These features, along with the dynamic typing and dynamic binding together make Python attractive to development teams working in a Rapid Application Development (RAD). As Python is an extracted level above the C language [11], Python can get used as the glue that connects existing components, as well as being able to be used as a scripting language [12]. Python gets considered to be easy to learn the language due to its high readability and is recommended by many exam boards as the language to use for teaching Computer Science at GCSE and A-Level level [?]. Python's simple and easy to learn syntax emphasises on readability, which, as a result, reduces the cost of program maintenance [12].

Python gets compared to a lot of other languages. However, due to the requirements and expectations of the application, we will compare it to other similar style applications that can potentially do a similar job. These being Java, JavaScript and C++. In general, the choice of the programming language to use is many other real-world constraints, for example, financial cost, availability, training and even personal preferences and attachments. However, we will focus on language issues for the comparisons.

In comparison to Java, Python programs will typically take 3-5 times quicker (See fig: 4.1) to develop but will have a slower run time. The time difference gets attributed to Python's built-in data types and its dynamic typing [13]. As Java gets better characterised as a low-level implementation language, this would be the language of choice if application execution speed was the deciding factor. If this is not a factor, then there is no real benefit over Python.

What gets said about Java is also the same when comparing C++ to Python. It is often 5-10 times shorter than equivalent C++ code! Anecdotal evidence suggests that one Python programmer can finish in two months what two C++ programmers cannot complete in a year. Python shines as a glue language, used to combine components written in C++ [13].

In comparison to JavaScript (JS), Python's 'object-based' subset is very similar to JS. Python supports a programming method that uses simple variables and functions, similar to JS, that do not need class definitions. However, Python also supports writing for much larger programs, which leads to more reusable code by using an accurate OOP way while with JS, that is all that it can do [13].

Another language that presented itself to us was the R language. R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues [14]. Many users think of R as a statistics system. We prefer to think of it as an environment within which statistical techniques get implemented. R can be extended (easily) via packages [14]. Academics and statisticians have developed R over two decades. R has now one of the richest ecosystems to perform data analysis. There are around 12000 packages available in CRAN (open-source repository). It is possible to find a library for whatever the analysis the user wants to perform. The wide variety of library makes R the first choice for statistical analysis, especially for specialised analytical work [15].

R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering) and graphical techniques, and is highly extensible. One of R's strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed [14]. The cutting-edge difference between R and the other statistical products is the output. R has fantastic tools to communicate the results. Rstudio comes with the library knitr. Xie Yihui wrote this package. He made reporting trivial and elegant. Communicating the findings with a presentation or a document is easy [15].

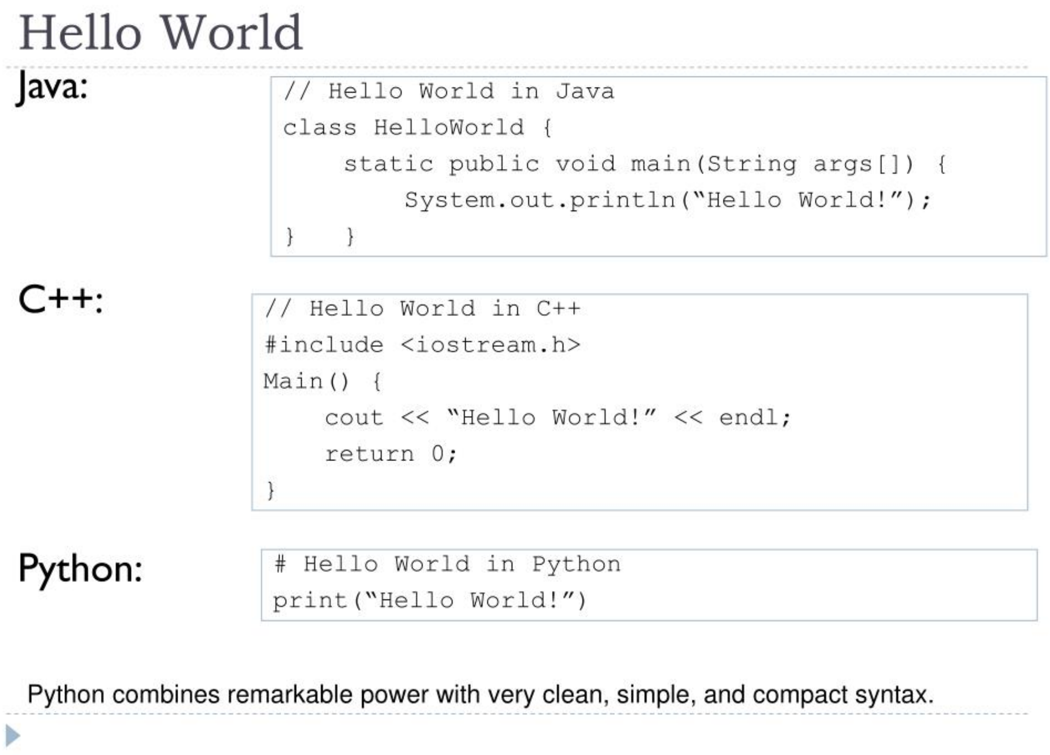
R is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form. It compiles and runs on a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux), Windows and macOS [14].

R, like S, is designed around a real computer language, and it allows users to add additional functionality by defining new functions. Much of the system is itself written in the R dialect of S, which makes it easy for users to follow the algorithmic choices made. For computationally-intensive tasks, C, C++ and Fortran code can be linked and called at run time. Advanced users can write C code to manipulate R objects directly [14].

R and Python are both open-source programming languages with a large community. New libraries or tools are added continuously to their respective catalogue. R is mainly used for statistical analysis, while Python provides a more general approach to data science. R and Python are both state of the art in terms of programming language oriented towards data science. Learning both of them is, of course, the ideal solution. R and Python requires a time-investment, and such luxury is not available for everyone. Python is a general-purpose language with a readable syntax. R, however, is built by statisticians and encompasses their specific language [15].

Python can pretty much do the same tasks as R: data wrangling, engineering, feature selection web scrapping, app and so on. Python is a tool to deploy and implement machine learning at a large-scale. Python codes are easier to maintain and more robust than R. Years ago; Python didn't have many data analysis and machine learning libraries.

Recently, Python is catching up and provides cutting-edge API for machine learning or Artificial Intelligence. Most of the data science job can be done with five Python libraries: Numpy, Pandas, Scipy, Scikit-learn and Seaborn [15].



Hello World

Java:

```
// Hello World in Java
class HelloWorld {
    static public void main(String args[]) {
        System.out.println(“Hello World!”);
}   }
```

C++:

```
// Hello World in C++
#include <iostream.h>
Main() {
    cout << “Hello World!” << endl;
    return 0;
}
```

Python:

```
# Hello World in Python
print(“Hello World!”)
```

Python combines remarkable power with very clean, simple, and compact syntax.

Figure 4.1: A comparison between Java, Python and C++ to print an output to the console. [1]

Python, on the other hand, makes replicability and accessibility easier than R. In fact, if you need to use the results of your analysis in an application or website, Python is the best choice [15].

In 2019 there was an active number of 26.66 billion devices attached to the internet [16,17], with an estimation of 35 billion in 2021 [16] and by 2025 75.44 billion [17]. Experts estimate that the IoT device market will reach $1.1 trillion in 2026 [16]. Every Second 127 new devices get connected to the world wide web [16].

With so many devices on the internet, an important consideration we had was to make the application web-based. By creating the application for the internet, this would allow potentially many more people to be able to access the application and interact with the different ML models.

In 1989 a British scientist named Sir Tim Berners-Lee started to work on something that would become to be known as the worldwide web (WWW) [18]. In October of 1990, Tim had created the three fundamental technologies required for the WWW to work [18]. These were HTML (HyperText Markup Language), which is the markup language for the web, URI (Uniform Resource Identifier), which is the unique address of where the information can get located and HTTP (HyperText Transfer Protocol). The HTTP allows for the retrieval of linked resources from over the www. CSS, on the other hand, was created in 1994 by Håkon Wium Lie at CERN. CSS got created due to believed need that HTML style sheet language for the web, to create "newspaper-like layout in a Web page" [19]. These two languages become the cornerstone of web pages. That was, until 1995, where JavaScript got introduced. Whereas HTML and CSS are used to give any website its structure and style, JavaScript is used to to add functionality and different behaviours to a website. It was, therefore, allowing users to interact with the website's content in many imaginative ways.

JS gets regarded as more of the language of the world-wide-web. It got initially designed to be used client-side in a web browser. However, it has in more recent years started to branch out and be able to be used to create applications on, not only the front end of the web but also desktops, servers and mobile platforms natively. For example, React Native, Node.js and TypeScript.

Since the release of JavaScript, it has surpassed Java, Flash, and other languages because it is relatively easy to learn, has a free and open community. JavaScript is also incredibly useful, allowing developers to be able to create apps with audiences in the millions quickly [20].

The decision on what language to use we a close call between Python and JavaScript, this was due to the massive amounts of libraries that were on offer and the support communities that were in place. With both being open source and both having essential libraries available to interact with machine learning models and visualisation tools, both could have been a perfect fit for the intended application. However, we decided upon using Python. Python was chosen based on it being the go-to language for anything machine learning related, and its ability to be able to be used multiplatform on desktops or mobile devices. Python supports modules and packages, which encourages programs to be developed modularity and therefore allows code to get reused. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms and can get freely distributed [12]. There was also an additional factor that the author was more familiar with Python and its required libraries compared to the libraries that will get required for using JavaSript.

There was the additional decision to use HTML and CSS within a small part of the project, the 'Learning Zone', based on the quickness of being able to create and host the webpages containing the learning content. It was allowing the learning content to evolve without having any impact on the overall development of the main application, allowing the learning content to be an individual entity within the main application.

4.2 Frameworks

4.2.1 GUI Framework

With the nature of the application, we needed to make the application have a Graphical User Interface (GUI). Having a GUI allowed the learning to be a lot more hands-on and allow the players to see what is happening within the models, especially when they interact with them.

Therefore, due to the GUI requirement, three GUI libraries presented themself to us. These were Pygame, PyQT5 and Tkinter.

Pygame is a free, open-sourced library. Released under the LGPL licence, Pygame is a set of Python modules designed for writing video games. Pygame adds functionality on top of the standard Python library. Pygame allows the user to create fully featured games and multimedia programs in the python language [21].

Pygame is highly portable and runs on nearly every platform and operating system, and it gets downloaded millions of times [21].

With the main aim of the application to be a game, Pygame was a strong contender. It was providing modules that can handle a lot of the key gaming mechanics and multiple screen switching. However, it lacked some key features that were deemed essential for the application. It was unable to provide a library that could create interactable graphs to be used as data inputs for the models and be able to render HTML and CSS content for the Learning Zone. Therefore reducing the amount of flexibility, it got decided upon for using HTML and CSS for the learning content. Therefore, meaning that all the content would need to be hardcoded. If any changes were needed, a significant transformation would need to happen to the overall code, instead of just changing the web content.

PyQt is a set of Python v2 and v3 bindings for The Qt Company's Qt application framework and runs on all platforms supported by Qt including Windows, macOS, Linux, iOS and Android. PyQt5 supports Qt v5. PyQt4 supports Qt v4 and will build against Qt v5. The bindings are implemented as a set of Python modules and contain over 1,000 classes [22].

PyQt brings together the Qt C++ cross-platform application framework and the cross-platform interpreted language Python. Qt is more than a GUI toolkit. It includes abstractions of network sockets, threads, Unicode, regular expressions, SQL databases, SVG, OpenGL, XML, a fully functional web browser, a help system, a multimedia framework, as well as a rich collection of GUI widgets. Qt classes employ a signal/slot mechanism for communicating between objects that is type safe but loosely coupled making it easy to create re-usable software components [22].

Qt also includes Qt Designer, a graphical user interface designer. PyQt is able to generate Python code from Qt Designer. It is also possible to add new GUI controls written in Python to Qt Designer [22].

PyQt combines all the advantages of Qt and Python. A programmer has all the power of Qt but can exploit it with the simplicity of Python [22].

Tkinter is the third option. Tkinter commonly comes bundled with Python, using Tk and is Python's standard GUI framework. It is famous for its simplicity and graphical user interface. It is open-source and available under the Python License [23].

Tkinter is Python's de-facto standard GUI (Graphical User Interface) package. It is a thin object-oriented layer on top of Tcl/Tk. Tkinter is not the only GuiProgramming toolkit for Python. It is however the most commonly used one. CameronLaird calls the yearly decision to keep TkInter "one of the minor traditions of the Python world [24]."

Tkinter supports functionality with Matplotlib, with Matplotlib offering libraries to allow handling the backend of the graph creation interacting with the GUI library. However, unlike QT, Tkinter does not support any GUI designer. Therefore the GUIs will have to be created programmatically, which will give more control, might involve more of a learning curve and potentially more time to implement in the initial stages.

After reviewing the different GUI libraries, PyQt was the decided library to use. We believed it would give us the ability to have

4.3 Packages Used

4.4 IDE Used

Pycharm

PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development [25].

VS Code

Visual Studio Code is a free source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging. First and foremost, it is an editor that gets out of your way. The delightfully frictionless edit-build-debug cycle means less time fiddling with your environment, and more time executing on your ideas [26].

At its heart, Visual Studio Code features a lightning fast source code editor, perfect for day-to-day use. With support for hundreds of languages, VS Code helps you be instantly productive with syntax highlighting, bracket-matching, auto-indentation, box-selection, snippets, and more. Intuitive keyboard shortcuts, easy customisation and community-contributed keyboard shortcut mappings let you navigate your code with ease.

For serious coding, you'll often benefit from tools with more code understanding than just blocks of text. Visual Studio Code includes built-in support for IntelliSense code completion, rich semantic code understanding and navigation, and code refactoring.

And when the coding gets tough, the tough get debugging. Debugging is often the one feature that developers miss most in a leaner coding experience, so we made it happen. Visual Studio Code includes an interactive debugger, so you can step through source code, inspect variables, view call stacks, and execute commands in the console. VS Code also integrates with build and scripting tools to perform common tasks making everyday workflows faster. VS Code has support for Git so you can work with source control without leaving the editor including viewing pending changes diffs [26]..

Customise every feature to your liking and install any number of third-party extensions. While most scenarios work "out of the box" with no configuration, VS Code also grows with you, and we encourage you to optimise your experience to suit your unique needs. VS Code is an open-source project so you can also contribute to the growing and vibrant community on GitHub.

Atom

Atom was developed and released by GitHub. This free, open-source code editor is self-labeled 'a hackable text editor for the 21st century'. And hackable, it is. Atom allows developers to fully customise the look, feel, and requirements to speed up their workflows. However, Atom still allows developers to use it productively without ever touching a config file. A freshly downloaded version comes pre-loaded with eight syntax themes and four UI: two light and two dark. But, if none of the pre-installed themes interest you, Atom makes it easy and quick to install customised themes created by a third-party or to create one yourself [27].

Chosen IDE VS Code - Due to familiarity and Extension features

4.5 Intricacies of the Game Components

4.5.1 Gameplay Area

4.5.2 Free Play Area

4.5.3 Learning Zone Area

4.5.4 Achievements Area

4.6 Example user stories (A UML term for case studies or example playthroughs)

Chapter 5

Conclusions and Future Work

In this document we have demonstrated the use of a LaTeX thesis template which can produce a professional looking academic document.

5.1 Contributions

The main contributions of this work can be summarized as follows:

- **A LaTeX thesis template**

  Modify this document by adding additional top level content chapters. These descriptions should take a more retrospective tone as you include summary of performance or viability.

- **A typesetting guide of useful primitive elements**

  Use the building blocks within this template to typeset each part of your document. Aim to use simple and reusable elements to keep your document neat and consistently styled throughout.

- **A review of how to find and cite external resources**

  We review techniques and resources for finding and properly citing resources from the prior academic literature and from online resources.

5.2 Future Work

Future editions of this template may include additional references to Futurama.

Bibliography

[1] Things Tech, "Which programming language to start with as a beginner," 2020, [Online; accessed August 10th, 2020]. [Online]. Available: https://thingsteck.wordpress.com/

[2] Internet Live Stats. (2020). [Online]. Available: https://www.internetlivestats.com

[3] G. Branwen. (2020) Internet search tips. [Online]. Available: https://www.gwern.net/Search

[4] RELX Group. (2019) Mendeley. [Online]. Available: https://www.mendeley.com

[5] J. T. Kajiya, "The rendering equation," in *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '86. New York, NY, USA: ACM, 1986, pp. 143–150.

[6] Apple Inc. (2020) Swift. [Online]. Available: https://developer.apple.com/documentation/swift

[7] M. Potuck. (2020) Apple hits 1.5 billion active devices with 80iphones and ipads running ios 13. [Online]. Available: https://9to5mac.com/2020/01/28/apple-hits-1-5-billion-active-devices-with-80-of-recent-iphones-and-ipads-running-ios-13/

[8] K. Finley. (2020) Python is more popular than ever. [Online]. Available: https://www.wired.com/story/python-language-more-popular-than-ever/

[9] B. Popper. (2020) The 2020 developer survey results are here! [Online]. Available: https://stackoverflow.blog/2020/05/27/2020-stack-overflow-developer-survey-results/

[10] A. Goel. (2020) Best programming language to learn in 2020 (for job and future). [Online]. Available: https://hackr.io/blog/best-programming-languages-to-learn-2020-jobs-future

[11] Stack Overflow. (2013) Python vs cpythony. [Online]. Available: https://stackoverflow.com/questions/17130975/python-vs-cpython

[12] Python Software Foundation. (2020) What is python? executive summary. [Online]. Available: https://www.python.org/doc/essays/blurb/

[13] ——. (2020) Comparing python to other languages. [Online]. Available: https://www.python.org/doc/essays/comparisons/

[14] The R Foundation. (2020) What is r? [Online]. Available: https://www.r-project.org/about.html

[15] Guru 99. (2020) R vs python: What's the difference? [Online]. Available: https://www.guru99.com/r-vs-python.html#:~:text=New%20libraries%20or%20tools%20are,general%20approach%20to%20data%20science.&text=Python%20is%20a%20general%2Dpurpose,and%20encompasses%20their%20specific%20language.

[16] G. D. Maayan. (2020) The iot rundown for 2020: Stats, risks, and solutions. [Online]. Available: https://securitytoday.com/Articles/2020/01/13/The-IoT-Rundown-for-2020.aspx?Page=1

[17] Statista. (2020) Internet of things (iot) connected devices installed base worldwide from 2015 to 2025. [Online]. Available: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

[18] World Wide Web Foundation. (2020) History of the web. [Online]. Available: https://webfoundation.org/about/vision/history-of-the-web/

[19] W3C. (2016) A brief history of css until 2016. [Online]. Available: https://www.w3.org/Style/CSS20/history.html

[20] T. DeGroat. (2019) The history of javascript: Everything you need to know. [Online]. Available: https://www.springboard.com/blog/history-of-javascript/

[21] Pygame. (2020) About. [Online]. Available: https://www.pygame.org/wiki/about

[22] Riverbank Computing. (2020) What is pyqt? [Online]. Available: https://riverbankcomputing.com/software/pyqt/intro

[23] A. Sharma. (2020) Introduction to gui with tkinter in python. [Online]. Available: https://www.datacamp.com/community/tutorials/gui-tkinter-python?utm_source=adwords_ppc&utm_campaignid=898687156&utm_adgroupid=48947256715&utm_device=c&utm_keyword=&utm_matchtype=b&utm_network=g&utm_adpostion=&utm_creative=229765585186&utm_targetid=aud-299261629574:dsa-429603003980&utm_loc_interest_ms=&utm_loc_physical_ms=1007460&gclid=Cj0KCQjwsuP5BRCoARIsAPtX_wGxaCEuqKDVDDyVASqiCw2zIKYwN2Duo3DObWGcfwQAjH3oxWK5WfoaAhafEALw_wcB

[24] Python Software Foundation. (2020) Tkinter. [Online]. Available: https://wiki.python.org/moin/TkInter

[25] PyCharm. (2020) Get started. [Online]. Available: https://www.jetbrains.com/help/pycharm/quick-start-guide.html

[26] Microsoft. (2020) Why did we build visual studio code? [Online]. Available: https://code.visualstudio.com/docs/editor/whyvscode

[27] CloudApp. (2020) A guide to atom text editor. [Online]. Available: https://www.getcloudapp.com/blog/how-to-use-atom-text-editor

Appendix A

Implementation of a Relevant Algorithm

```c
#include <stdio.h>

int main(int argc, char *argv[]) {
  printf("Hello world.\n");
  return 0;
}
```

Listing A.1: An implementation of an important algorithm from our work.

Appendix B

Supplementary Data

The results of large ablative studies can often take up a lot of space, even with neat visualization and formatting. Consider putting full results in an appendix chapter and showing excerpts of interesting results in your chapters with detailed analysis. You can use labels and references to refer the reader here for the full data.